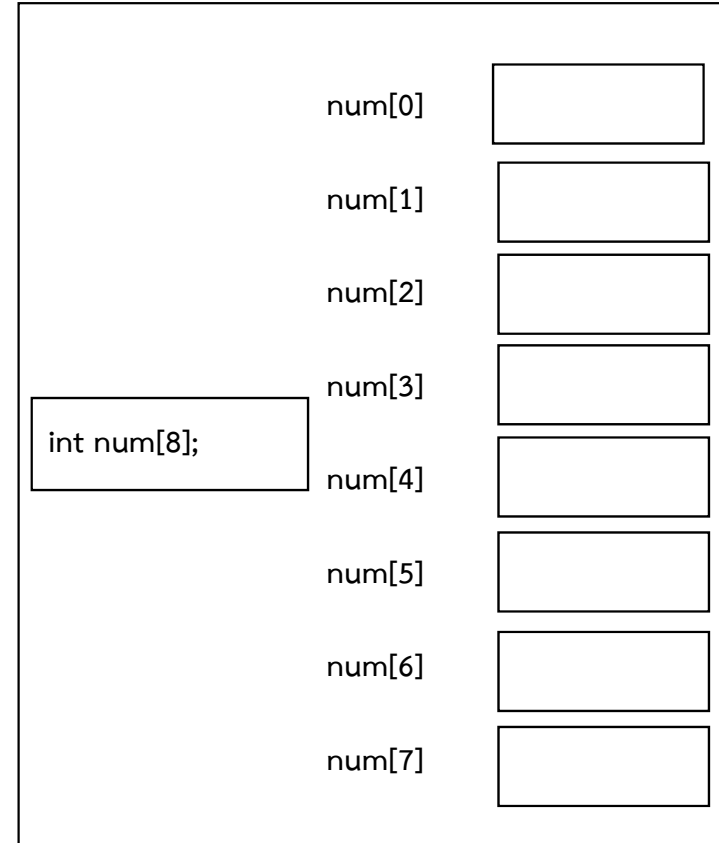
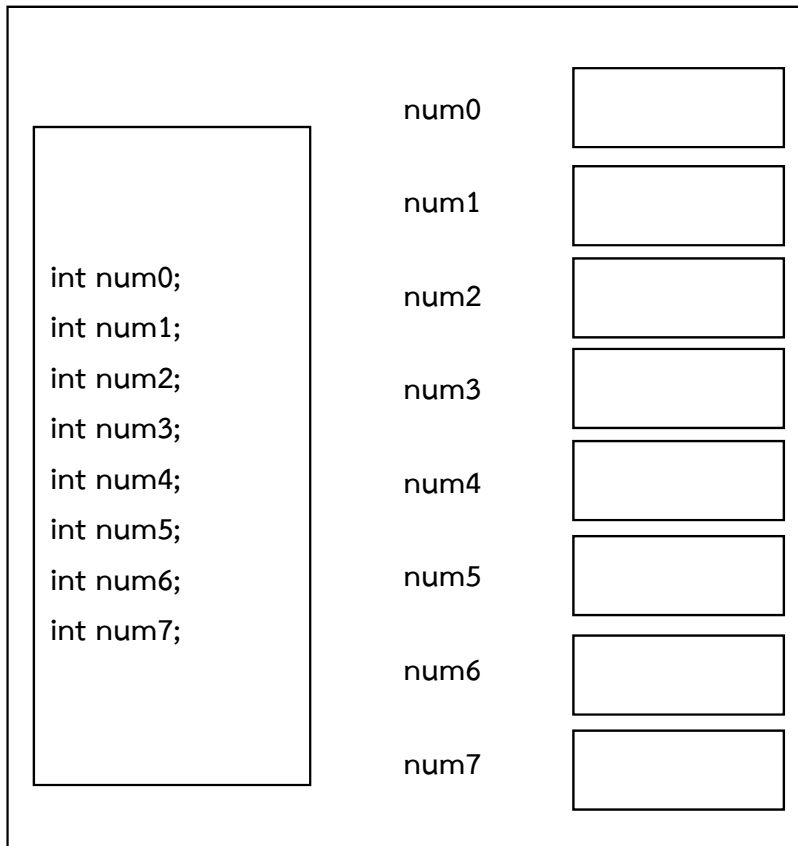


อาร์เรย์ (Array)

ค่ายโอลิมปิกวิชาการ ครั้งที่ 22 ปีการศึกษา 2564 สาขาคอมพิวเตอร์ ค่ายที่ 1

23 มีนาคม 2565

การเปรียบเทียบตัวแปรชนิดข้อมูลพื้นฐานกับอาร์เรย์ ในการเก็บค่าจำนวนเต็ม



การประกาศตัวแปร (Array declaration)

รูปแบบ

```
type  arrayName[arraySize]
```

type	คือ ชนิดข้อมูลที่จะเก็บไว้ในตัวแปรอาร์เรย์
arrayName	คือ ชื่อตัวแปรอาร์เรย์
arraySize	คือ ขนาดของอาร์เรย์ หมายถึงจำนวนสูงสุดของข้อมูลที่สามารถเก็บไว้ในอาร์เรย์

การประกาศตัวแปร (Array declaration)

```
int num[5];
```

num[0] num[1] num[2] num[3] num[4]



```
float data[10];
```

data[0] data[1] data[2] data[3] data[4] data[5] data[6] data[7] data[8] data[9]



- ตำแหน่งแรกในอาร์เรย์ มีค่า **index = 0**
- ถ้าอาร์เรย์มีขนาด **n** ตำแหน่งสุดท้ายในอาร์เรย์มีค่า **index = n-1**
- ค่าตำแหน่งที่อยู่ในอาร์เรย์จะเรียงต่อกัน เช่น ถ้า data[0] อยู่ที่ตำแหน่งในหน่วยความจำ 2120d แล้ว data[1] จะอยู่ที่ตำแหน่ง 2124d เนื่องจาก float ใช้พื้นที่ในหน่วยความจำ 4 bytes

การกำหนดค่าเริ่มต้น (Array Initialization)

```
type  arrayName[arraySize] = {value1 , value2 , ... } ;
```

value1 , value2 , ... คือ ค่าเริ่มต้นที่กำหนดให้กับอาร์เรย์ตรงตามชนิดข้อมูลที่กำหนด

ในกรณีที่กำหนดค่าเริ่มต้นในอาร์เรย์ไม่ครบตามจำนวนของอาร์เรย์ ค่าของตำแหน่งที่เหลือจะถูกกำหนดให้เป็นศูนย์

```
int num[5] = {2,10,3,11,5};
```

กำหนดค่าเริ่มต้นครบตามขนาดของอาร์เรย์

2	10	3	11	5
---	----	---	----	---

```
int num[5] = {2,10,3};
```

กำหนดค่าเริ่มต้นบางค่า ค่าที่ไม่กำหนดจะมีค่าเป็นศูนย์

2	10	3	0	0
---	----	---	---	---

```
int num[5] = {0};
```

กำหนดทุกค่าในอาร์เรย์เป็นศูนย์

0	0	0	0	0
---	---	---	---	---

```
int num[] = {2,10,3,11,5};
```

กำหนดค่าเริ่มต้นโดยไม่กำหนดขนาดอาร์เรย์ แต่ขนาดของอาร์เรย์เป็นตามจำนวนของค่าที่กำหนด

2	10	3	11	5
---	----	---	----	---

การเข้าถึงสมาชิกในอาร์เรย์ (Accessing Array)

```
arrayName[index]
```

arrayName คือ ชื่อตัวแปรอาร์เรย์

index คือ ค่าดัชนีกำกับ

เพื่ออ้างตำแหน่งข้อมูลในอาร์เรย์ที่ต้องการเข้าถึง

```
printf("%d",num[3]);
```

เช่น ต้องการแสดงค่าในอาร์เรย์ตำแหน่งที่ 3 ชื่อตัวแปร num ซึ่งเก็บค่าจำนวนเต็ม

การกำหนดค่าในอาร์เรย์ (Assigning Values to Arrays)

รูปแบบ

```
arrayName[index] = expression;
```

ตัวอย่าง

```
num[3]=5;
```

```
num[6]=num[3]*10;
```


การกำหนดค่าในอาร์เรย์ (Assigning Values to Arrays)

การคัดลอกข้อมูลในอาร์เรย์ชื่อ one ให้กับตัวแปรอาร์เรย์ชื่อ two ซึ่งมีขนาดเท่ากับ 20 ทำได้ดังนี้

```
for(i=0; i<20; i++)  
    two[i]=one[i];
```

การรับข้อมูลเข้าเก็บในอาร์เรย์ (Entering Values in Arrays)

ใช้คำสั่ง `scanf` รับค่าเก็บไว้ในอาร์เรย์ชื่อ `num` ในตำแหน่งที่ 5

```
scanf("%d", &num[5]);
```

การรับค่าเก็บไว้ในอาร์เรย์ทุกตำแหน่งชื่อ `num` ขนาด 20 ใช้คำสั่ง
วนลูปทุกตำแหน่ง ในที่นี้ใช้ `for`

```
for(i=0; i<20; i++)  
    scanf("%d", &num[i]);
```

การรับค่าคะแนนสอบกลางภาคของนักเรียนจำนวน 20 คนเก็บในตัวแปรอาร์เรย์

```
1. #include<stdio.h>
2. int main()
3. {
4.     int score[20];
5.     int i;
6.
7.     for(i=0;i<20;i++){
8.         printf("Enter score %d : ",i+1);
9.         scanf("%d",&score[i]);
10.    }
11.    return 0;
12.}
```

ผลลัพธ์

Enter score 1 : 20

Enter score 2 : 52

Enter score 3 : 52

.

.

.

Enter score 18 : 51

Enter score 19 : 61

Enter score 20 : 71

การแสดงค่าในอาร์เรย์ (printing Values in Arrays)

ใช้คำสั่ง printf แสดงค่าในอาร์เรย์ชื่อ num ในตำแหน่งที่ 5

```
printf("%d", num[5]);
```

การแสดงค่าในอาร์เรย์ทุกตำแหน่งชื่อ num ขนาด 20 ใช้คำสั่งวน
ลูปทุกตำแหน่ง ในที่นี้ใช้ for

```
for(i=0; i<20; i++)  
    printf("%d", num[i]);
```

```

1. #include<stdio.h>
2. int main()
3. {
4.     int score[20];
5.     int i;
6.
7.     for(i=0;i<20;i++){
8.         printf("Enter score %d : ",i+1);
9.         scanf("%d",&score[i]);
10.    }
11.    for(i=0;i<20;i++)
12.        printf("score %d : %d\n",i+1,score[i]);
13.    return 0;
14.    }

```

ผลลัพธ์

```

Enter score 1 : 20
Enter score 2 : 52
Enter score 3 : 52
.
.
.
Enter score 18 : 51
Enter score 19 : 61
Enter score 20 : 71
score 1 : 20
score 2 : 52
score 3 : 52
.
.
.
score 18 : 51
score 19 : 61
score 20 : 71

```

โปรแกรมแสดงค่าในตัวแปรอาร์เรย์ชื่อ num

```
1. #include<stdio.h>
2. int main()
3. {
4.     int num[15]={4,5,3,29,10,12,6,8,
5.                 9,11,15,14,7,19,17};
6.     int i;
7.     for(i=0; i<15; i++){
8.         printf("%d ", num[i]);
9.     }
10.    return 0;
11. }
```

ผลลัพธ์

4 5 3 29 10 12 6 8 9 11 15 14 7 19 17

โปรแกรมแสดงค่าในตัวแปรอาร์เรย์แถวละ 5 ค่า

```
1. #include<stdio.h>
2. #define SIZE 20
3. #define NUM_PER_ROW 5
4. int main()
5. {
6.     int num[SIZE]={4,5,3,29,10,12,6,8,9,11,
7.                    15,14,7,19,17,1,2,18,19,20};
8.     int i;
9.     int j;
10.    for(i=0; i< SIZE/NUM_PER_ROW ; i++){
11.        for(j=0; j< NUM_PER_ROW ; j++){
12.            printf("%d ",num[NUM_PER_ROW*i+j]);
13.        }
14.        printf("\n");
15.    }
16.    return 0;
17. }
```

ผลลัพธ์

```
4 5 3 29 10
12 6 8 9 11
15 14 7 19 17
1 2 18 19 20
```

โปรแกรมหาผลรวมของค่าในอาร์เรย์

```
1. #include<stdio.h>
2. #define SIZE 10
3. int main()
4. {
5.     int numbers[SIZE] = {23,10,40,30,78,42,10,34,17,30};
6.     int i;
7.     int sum=0;
8.     for(i=0; i< SIZE ; i++)
9.         sum = sum + numbers[i];
10.    printf("sum = %d\n",sum);
11.    return 0;
12.}
```

ผลลัพธ์

sum = 314

โปรแกรมหาค่าเฉลี่ยของคะแนนสอบของนักเรียน 10 คน

```
1. #include<stdio.h>
2. #define SIZE 10
3. int main()
4. {
5.     float score[SIZE];
6.     int i;
7.     float sum = 0;
8.     printf("Please enter students\' score : \n");
9.     for(i=0; i< SIZE ; i++){
10.         printf("%d : ",i+1);
11.         scanf("%f",&score[i]);
12.     }
13.     for(i=0; i< SIZE ; i++)
14.         sum += score[i];
15.     printf("Average score = %.2f\n",sum/SIZE);
16.     return 0;
17. }
```

ผลลัพธ์

enter students' score :

1 : 25.6

2 : 27.4

3 : 22.3

4 : 21.5

5 : 20.3

6 : 19.8

7 : 28.6

8 : 24.6

9 : 27.6

10 : 24.3

Average score = 24.20

โปรแกรมสร้างอาร์เรย์สองอาร์เรย์แทนกลุ่มของจำนวนเต็มคู่ และจำนวนเต็มคี่

```
1. #include<stdio.h>
2. #define SIZE 10
3. int main()
4. {
5.     int numbers[SIZE] ;
6.     int even[SIZE] ;
7.     int odd[SIZE] ;
8.     int i;
9.     int j=0;
10.    int k=0;
11.    printf("Enter 10 integer:\n");
12.    for(i=0;i< SIZE ;i++){
13.        printf("%d : ",i+1);
14.        scanf("%d",&numbers[i]);
15.    }
16.    for(i=0;i< SIZE ;i++){
17.        if(numbers[i] % 2 == 0){
18.            even[j] = numbers[i];
19.            j++;
20.        }
21.        else{
22.            odd[k] = numbers[i];
23.            k++;
24.        }
25.    }
26.    printf("\nEven numbers : \n");
27.    for(i=0;i< j ;i++){
28.        printf("%d ", even[i]);
29.    }
30.    printf("\nOdd numbers : \n");
31.    for(i=0;i< k ;i++){
32.        printf("%d ", odd[i]);
33.    }
34.    return 0;
35. }
```

โปรแกรมหาค่าความถี่ของตัวเลข 1 ถึง 9 ที่อยู่ในอาร์เรย์

```
1. #include<stdio.h>
2. #define SIZE 10
3. int main()
4. {
5.     int numbers[SIZE] ;
6.     int even[SIZE] ;
7.     int odd[SIZE] ;
8.     int i;
9.     int j=0;
10.    int k=0;
11.    printf("Enter 10 integer:\n");
12.    for(i=0;i< SIZE ;i++){
13.        printf("%d : ",i+1);
14.        scanf("%d",&numbers[i]);
15.    }
```

```
16.    for(i=0;i< SIZE ;i++){
17.        if(numbers[i] % 2 == 0){
18.            even[j] = numbers[i];
19.            j++;
20.        }
21.        else{
22.            odd[k] = numbers[i];
23.            k++;
24.        }
25.    }
26.    printf("\nEven numbers : \n");
27.    for(i=0;i< j ;i++){
28.        printf("%d ", even[i]);
29.    }
30.    printf("\nOdd numbers : \n");
31.    for(i=0;i< k ;i++){
32.        printf("%d ", odd[i]);
33.    }
34.    return 0;
35. }
```

ผลลัพธ์

Enter 10 integer:

1 : 11

2 : 14

3 : 12

4 : 15

5 : 17

6 : 13

7 : 16

8 : 19

9 : 11

10 : 10

Even number:

14 12 16 10

Odd number:

11 15 17 13 19 11

Quick Sum

- เป็นการหาผลบวกอย่างรวดเร็วภายในช่องอาร์เรย์ที่ติดกัน

ตัวอย่าง มีข้อมูลจำนวนประชากรของท้องที่ตามถนนสายหนึ่ง ที่ยาว 10 กิโลเมตร

5 8 9 2 0 7 4 3 0 8 (พันคนต่อตารางกิโลเมตร)

จงหาจำนวนประชากรสูงสุดบนช่วงถนนที่ติดกันยาว 5 กิโลเมตร

```
for (i=0 ; i<N-5 ; i++) {  
    sum=0 ;  
    for (j=i ; j<N+5 ; j++)  
        sum+=num[i] ;  
    if (sum>max_sum) max_sum = sum ;  
}
```

```

#include<stdio.h>
int main(){
    int N,i,j,max_sum=0,sum;
    scanf("%d",&N);
    int num[N];

    for(i=0;i<N;i++){
        scanf("%d",&num[i]);

    for(i=0;i<N-5;i++){
        sum=0;
        for(j=i;j<i+5;j++){
            sum+=num[j];
            if(sum>max_sum) max_sum = sum;
        }
    }
    printf("%d",max_sum);
    return 0;
}

```

10

5 8 9 2 0 7 4 3 0 8

26

1D Quick Sum

เอาผลลัพธ์เดิมมาใช้ในการบวกเข้าและลบออกแทน ทำได้ 2 รูปแบบ

5	8	9	2	0	7	4	3	0	8
---	---	---	---	---	---	---	---	---	---

- แบบที่ 1

หาผลบวก 5 ตัวแรกก่อน $\overset{\times}{5} + 8 + 9 + 2 + 0 = 24$

ชุดต่อไปคือ $8 + 9 + 2 + 0 + 7$ ซึ่ง $8 + 9 + 2 + 0$ เป็นงานซ้ำซ้อน

เราใช้ผลบวกเดิมลบด้วย 5 และบวกด้วย 7 เข้าไปจะได้ผลบวกชุดใหม่เป็น $24 - 5 + 7 = 26$

1D Quick Sum

เอาผลลัพธ์เดิมมาใช้ในการบวกเข้าและลบออกแทน ทำได้ 2 รูปแบบ

5	8	9	2	0	7	4	3	0	8
---	---	---	---	---	---	---	---	---	---

- แบบที่ 2

หาผลบวกจากจุดเริ่ม (เลข 5) ไปจนถึงจุดที่ i ทุกจุดเก็บไว้ก่อน

แบบนี้จะได้ผลลัพธ์ที่เก็บไว้เป็น 5 13 22 24 24 31 35 38 38 46


เดิม 5 8 9 2 0 7 4 3 0 8

จากนั้นนำค่าที่เก็บไว้มาลบกัน เช่น จะหาค่าของ $8+9+2+0+7$ มันคือ $31-5=26$